API Authorization

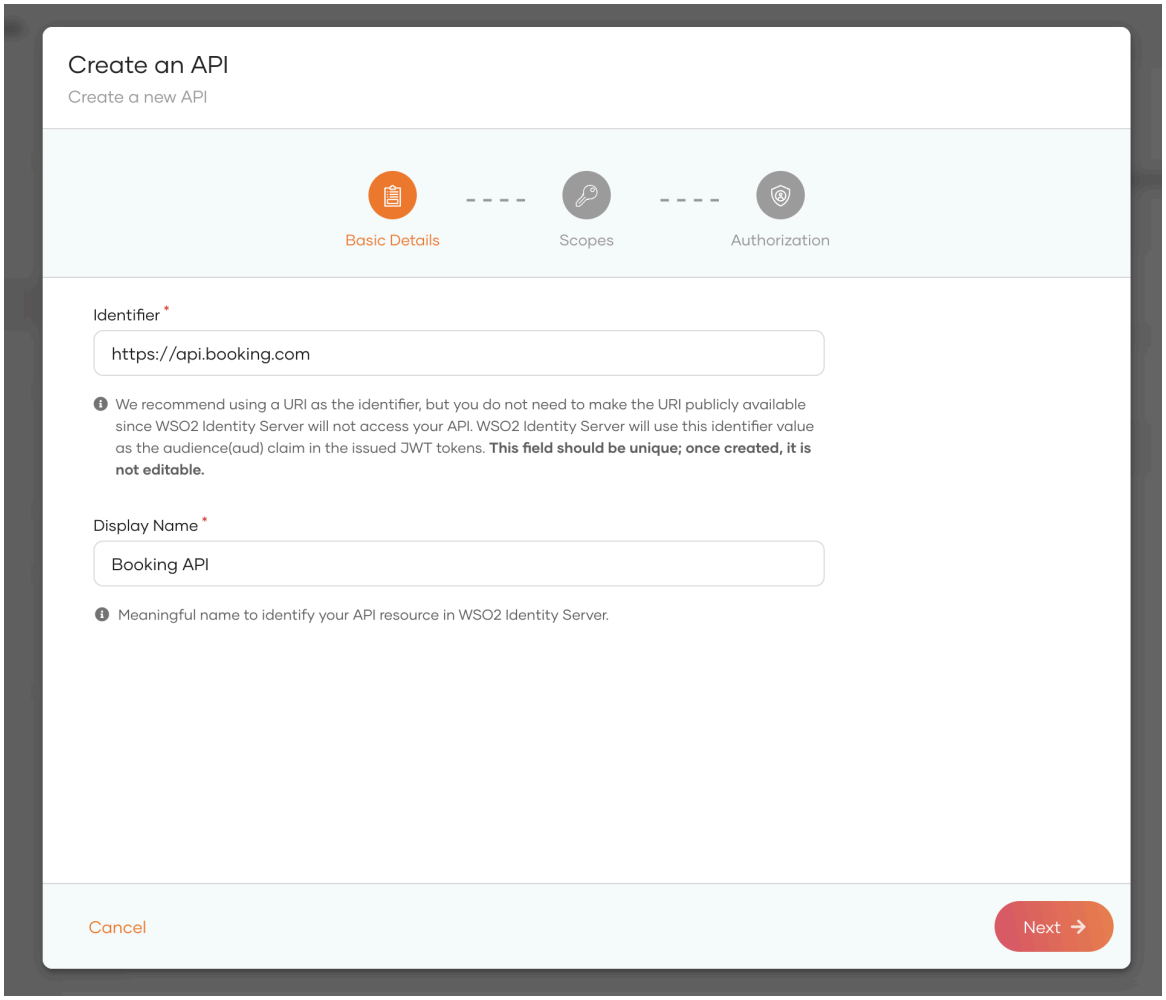
WSO2 Identity Server allows authorizing user access to an application's API resources based on the application associated roles assigned to the users or user groups. It also allows you to validate the scope of an OAuth access token using XACML policies to provide fine-grained access control to APIs. As the XACML is a deprecated feature, in this section we will discuss **Role Based Access Control**.

In RBAC what happens is, Identity Server validates the user's role assignment (direct or via groups), examines the permissions associated with the roles, and decides whether to permit or restrict the user's access to the API resources.

Register an API resource¶

1. On the WSO2 Identity Server Console, go to **API Resources**.
2. Click **+ New API** to register a new API resource.
3. Enter the following details:

Identifier	This is an identifier for your API resource. This can be any value, but WSO2 Identity Server recommends using the URI of the API resource as the identifier. This value will be used as the aud claim in the issued JWT token.
Display Name	A meaningful name to identify your API resource in WSO2 Identity Server.



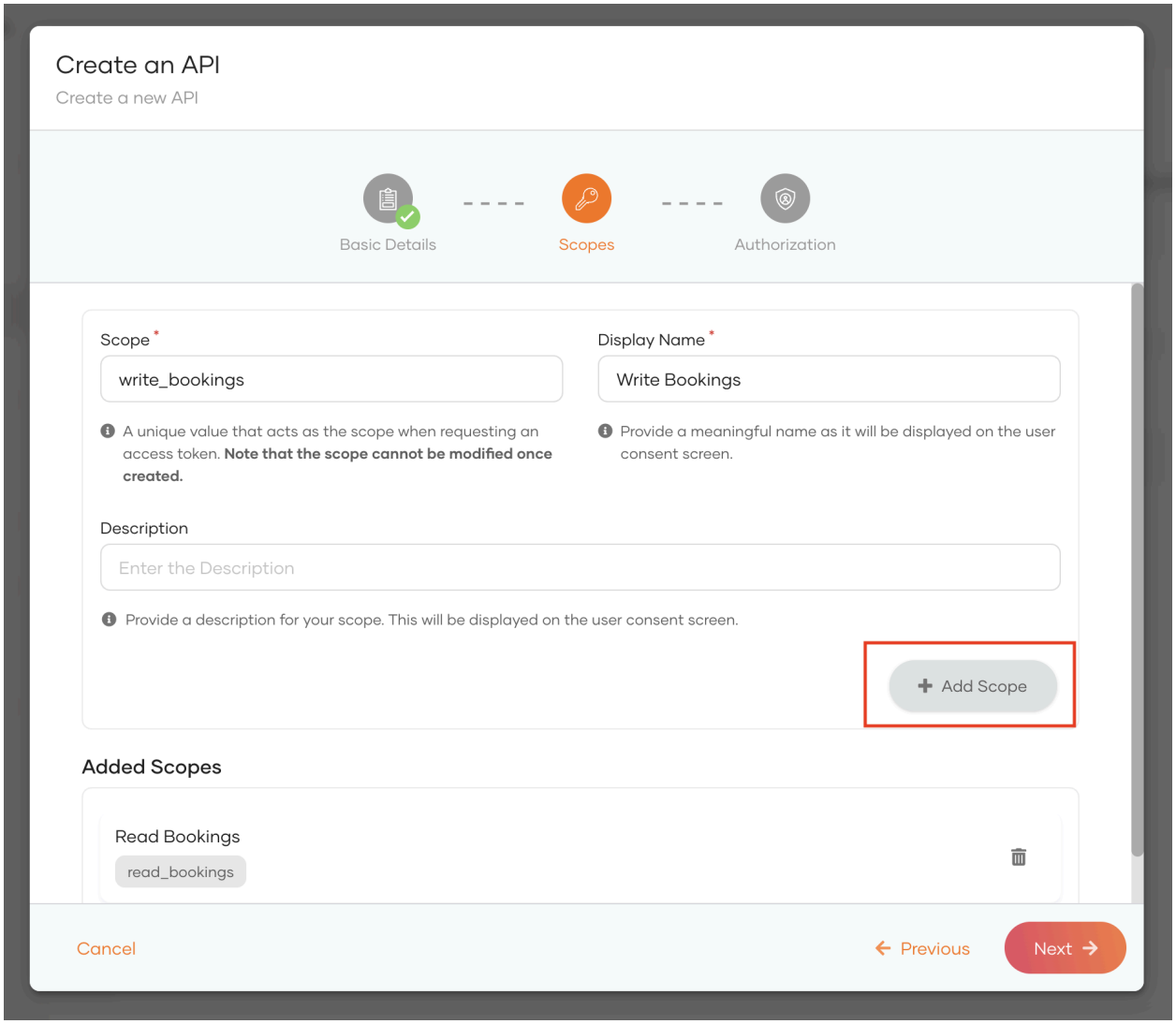
4. Click **Next.**
5. Add **Scope** and **Display Name** and click **+ Add Scope.**
6. Add the below two scope

    Scope: read_bookings

    Display Name: Read Bookings

    Scope: write_bookings

    Display Name: Write Bookings



7. Click **Next**.
8. Keep the **Requires Authorization** option checked and Click **Finish**.

Authorize the API resources for an app¶

Once you have registered API resources, you can authorize applications in your organization to access those API resources. This is done by connecting the API resources to the relevant applications.

Follow the below steps to authorize the API resource for an app.

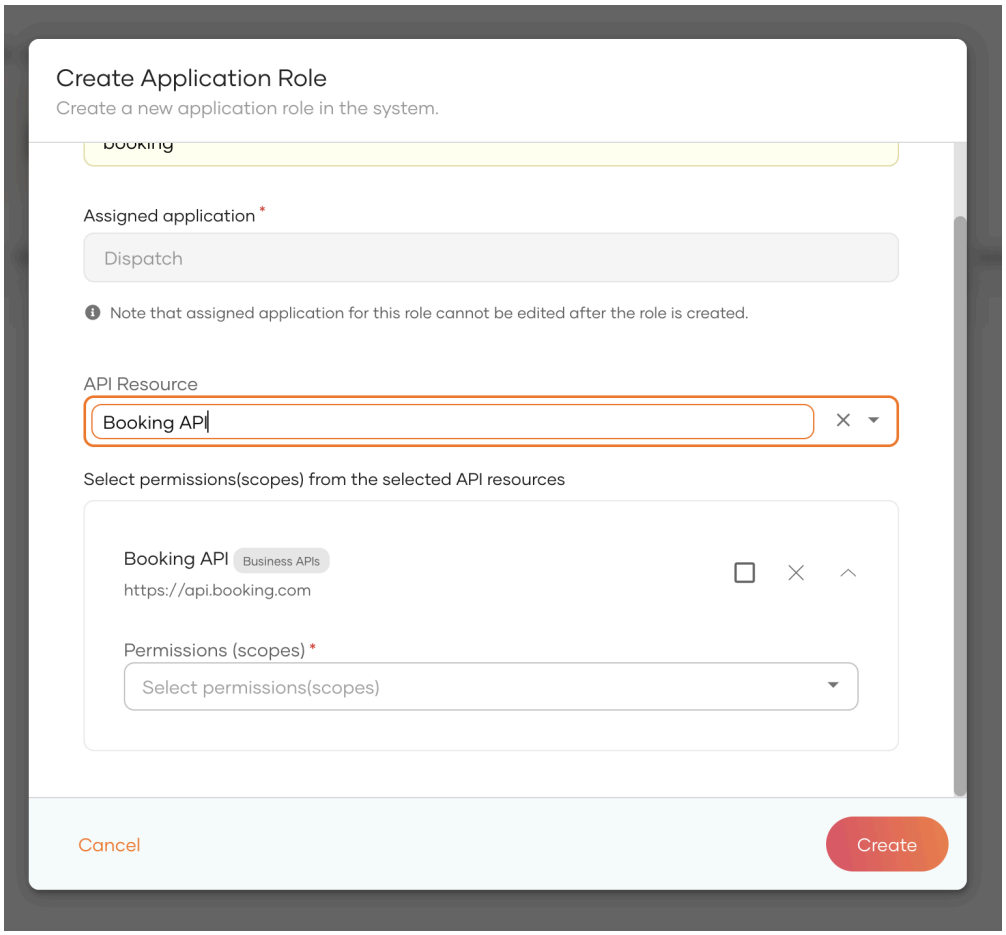
1. Use the sample app created in **6. Identity Federation** tutorial.

> **Warning**
>
> Note that you cannot authorize API resources for a SAML application.

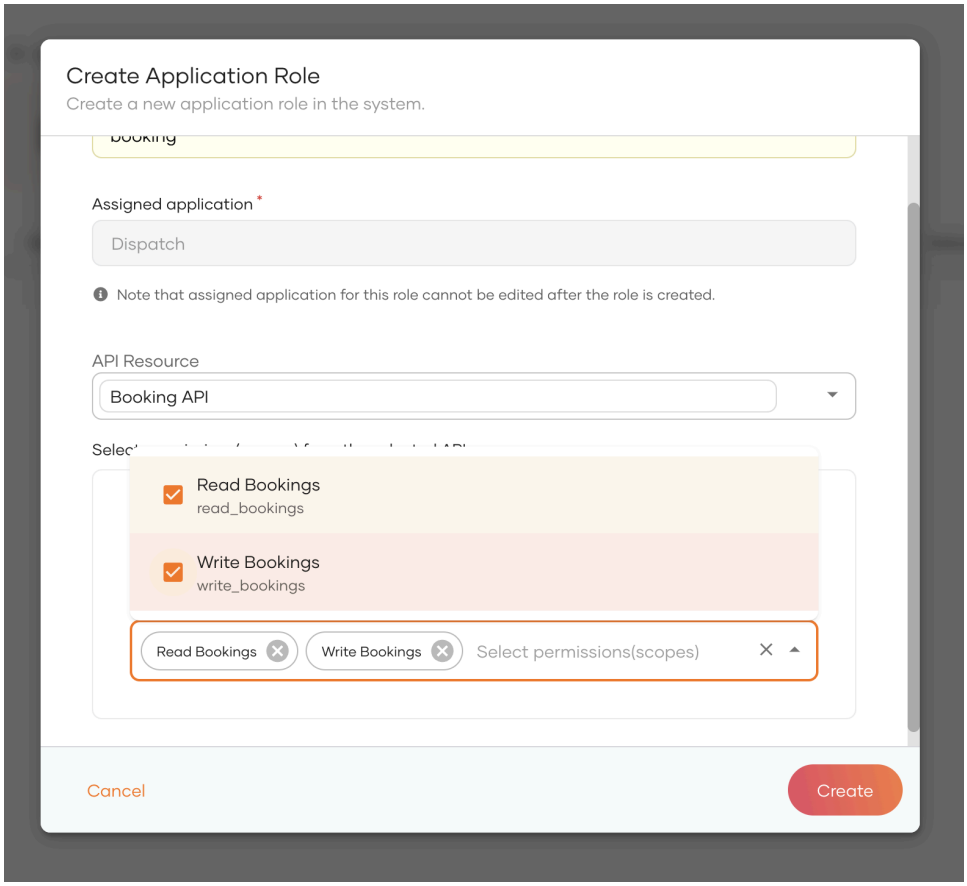
2. Enable login consent.
    a. View the created application and Goto **Advanced** tab.
    b. Uncheck **Skip login consent**.
    c. Click **Update**.
3. View the created application and goto **API Authorization** tab.
4. Click on **Authorize an API Resource**.
5. From the API Resource list select Booking API.
6. For Authorize scopes, select both scopes, **Read Bookings** and **Write Bookings**.
7. For the **Authorization Policy** since you have checked **Requires Authorization** scope while registering the API Resource, RBAC will be selected by default.
8. Click **Finish**.

Associate roles to the application

1. Create an Application Role and assign the scopes of the registered resources.
2. Go to select the Application **Dispatch**.
3. Goto **Roles** tab.
4. Clear any existing roles added.
5. Select Role Audience as Application and click on **+ New Role**.
6. Add **booking** as the role name.
7. As the API Resource, select **Booking API**.



8. Select **read_bookings** and **wrie_bookings** as permissions (scopes).

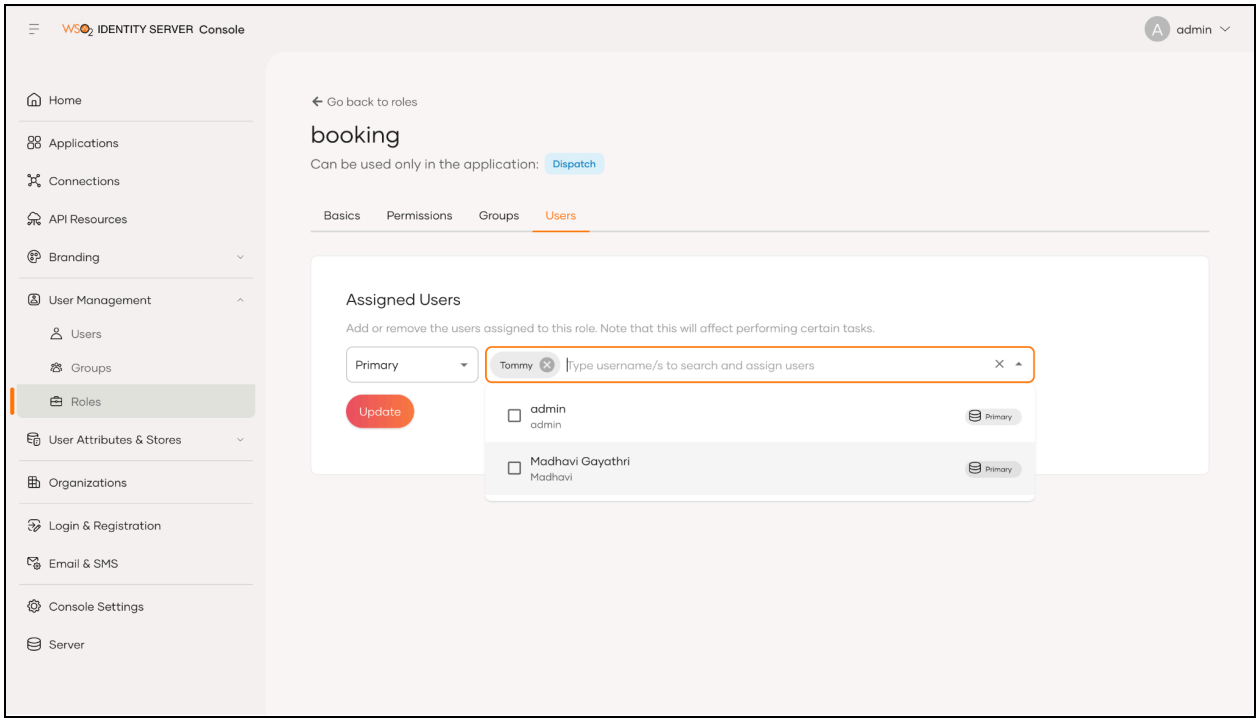


9. Click **Update**.

Assign users or groups to roles

Assign the created booking application role to the already created user Tommy.

1. On the WSO2 Identity Server Console, expand the **User Management** menu item and select **Roles**.
2. From the list of roles, select **booking**, application role.
3. Goto the **Users** tab.
4. Click on **Assign Users**.
5. Click on **Type username** area.
6. From the list of users select **Tommy** and click **Update**.

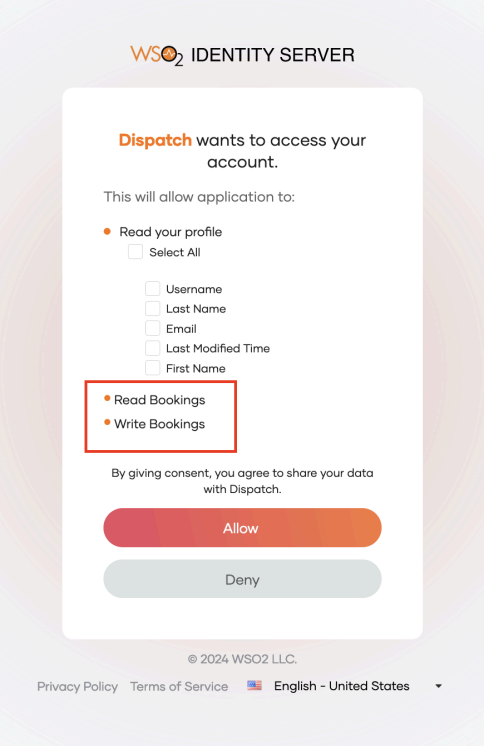


Request registered resource from the client app end

1. Open `<TOMCAT_PATH>/webapps/pickup-dispatch/WEB-INF/classes/dispatch.properties` file.
2. Add **read_bookings** and **write_bookings** as scope names.
3. Save the file and restart the Tomcat server.

Try it

1. Access the link http://localhost.com:8080/pickup-dispatch/oauth2client
2. and click on the **LOGIN** button.
3. Add the credentials of **Tommy** user and sign in.
4. You will be prompted to give permission to the below scopes.



5. Click **Allow**.
6. By consenting to the scopes you will be granting access to the client application to use the API resources associated with it.